

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 1° parte (6 CFU)

10 Gennaio 2013 – Tempo a disposizione: 2 h – Risultato: 32/32 punti

## Esercizio 1 (6 punti)

Si considerino le seguenti frasi:

- Tutte le madri amano le loro figlie
- Grimilde è una madre
- Biancaneve è figlia di Grimilde
- Se una madre ama sua figlia, non l'avvelena
- Grimilde avvelena Biancaneve

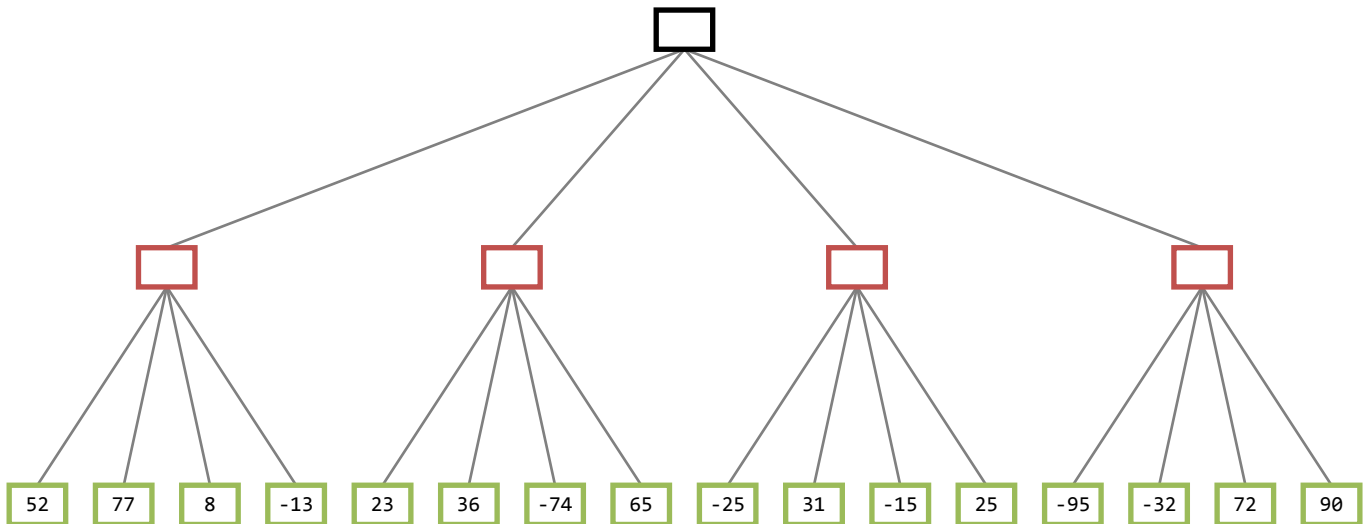
Si formalizzino in *logica dei predicati del I ordine*, utilizzando i seguenti predicati:

- madre(X) X è una madre
- ama(X, Y) X ama Y
- figlia(X, Y) X è figlia di Y
- avvelena(X, Y) X avvelena Y

Infine si trasformino in clausole e si dimostri, applicando la risoluzione, che la teoria ottenuta è inconsistente (la risoluzione deriva la clausola vuota, equivalente a *false*).

## Esercizio 2 (6 punti)

Si consideri il seguente albero di gioco in cui la valutazione dei nodi terminali è dal punto di vista del primo giocatore (*max*). Si mostri come gli algoritmi *min-max* e *alfa-beta* risolvono il problema.



## Esercizio 3 (6 punti)

Il seguente programma *Prolog* calcola se un giocatore (*min* o *max*) vince in un albero di gioco *min-max*, in cui le foglie sono etichettate con il nome del vincitore.

```
vince(X, X).
vince(P, tree(L, R)) :- changePlayer(P, NP), not(vince(NP, L)), !.
vince(P, tree(L, R)) :- changePlayer(P, NP), not(vince(NP, R)).
changePlayer(min, max).
changePlayer(max, min).
```

Si mostri l'*albero SLD-NF* relativo al goal:

```
?- vince(max, tree(min, max)).
```

#### Esercizio 4 (4 punti)

Si definisca un predicato *Prolog* ultimo/2 che, data una lista L di interi in ingresso, ne restituisca l'ultimo elemento. Se la lista è vuota, il predicato fallisce.

```
Es.: ?- ultimo([1, 2, 3], 3).  
      yes  
      ?- ultimo([1, 2, 3], X).  
      yes      X=3  
      ?- ultimo([], 3).  
      no
```

#### Esercizio 5 (7 punti)

Si hanno a disposizione due brocche  $b1$  e  $b2$  – una da 5 lt e una da 3 lt – e una sorgente d'acqua. Le brocche sono inizialmente vuote. Si vuole raggiungere uno stato in cui in una delle due brocche c'è 1 lt d'acqua e l'altra è vuota, attingendo la minor quantità d'acqua alla fonte e avendo a disposizione i seguenti operatori:

- OP1. Riempimento della brocca  $b1$  (se vuota), costo 5 lt
- OP2. Riempimento della brocca  $b2$  (se vuota), costo 3 lt
- OP3. Svuotamento della brocca  $b1$ , costo 0
- OP4. Svuotamento della brocca  $b2$ , costo 0
- OP5. Travaso da  $b1$  a  $b2$ , fino a riempire  $b2$  o a svuotare  $b1$ , costo pari ai lt travasati
- OP6. Travaso da  $b2$  a  $b1$ , fino a riempire  $b1$  o a svuotare  $b2$ , costo pari ai lt travasati

Si rappresenti lo stato con la coppia  $(B1, B2)$ , dove  $B1$  rappresenta la quantità d'acqua nella brocca  $b1$  di capacità 5 lt e  $B2$  quella di  $b2$  con capacità 3 lt. Si mostri come l'*algoritmo A\** trova la soluzione, considerando come costo di cammino quello ottenuto in base agli operatori applicati, e come euristica la distanza dall'obiettivo calcolata dalla seguente funzione:  $(B1+B2-1)$ , ovvero numero di lt contenuti in  $b1$  più quelli in  $b2$  meno 1 lt (per lo stato iniziale tale euristica da valore -1).

#### Esercizio 6 (3 punti)

Si descriva la tecnica di *Forward Checking (FC)* e il suo algoritmo (se lo si desidera portando un semplice esempio di applicazione). Si discutano differenze e analogie del *FC* rispetto alla tecnica di *Arc Consistency (AC)*.

**VOTO:**

- Esame da 6 CFU, il voto è determinato da questa I parte
- Esame da 9 CFU, è la media pesata della I parte (che vale 2/3) e della II (che vale 1/3) ovvero il voto finale è dato da:  $2*(votoParteI+votoParteII)/3$  e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte (3 CFU)

10 Gennaio 2013 – Tempo: 30 + 15min – Risultato: 16/16 punti

## Esercizio 7 (5 punti)

Si introduca brevemente il concetto di *Open World Assumption* (OWA) adottato nelle logiche descrittive, e si definisca un piccolo esempio di base di conoscenza, mostrando come a partire da tale conoscenza si possa giungere a conclusioni differenti a seconda che si adotti la OWA o la *Close World Assumption* (CWA).

## Esercizio 8 (8 punti)

Si scriva un meta-predicato:

```
trova_tutte(Goal, ListOut)
```

che, dato in ingresso un Goal con predicato binario, restituisce in ListOut (lista di uscita), le soluzioni (come coppie di elementi) derivanti dalla sua invocazione.

Ad esempio, supponendo di avere anche le clausole:

```
p(1, 2).  
P(2, 3).  
P(a, b).
```

l'invocazione seguente ha successo con esito come indicato:

```
?- trova_tutte(p(X, Y), ListOut).  
yes L=[(1, 2), (2, 3), (a, b)]
```

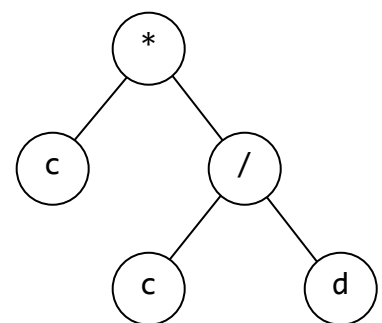
## Esercizio 9 (3 punti) – ulteriori 15 min

**Nota: lo deve svolgere solamente chi non ha partecipato alla lezione/esercitazione del 19 Novembre 2010 su Prolog e grammatiche.**

Data la seguente grammatica:

```
G = (Vn, Vt, P, S)  
Vn = {E, T}  
Vt = *, /, c, d}  
P = {E ::= T * E | T / E | T ::= c | d}  
S = E
```

Si scrivano le corrispondenti clausole in versione DCG estesa che verificano la correttezza di una frase e producono l'albero sintattico astratto. Ad esempio, per il goal  $c*c/d$  (il cui albero sintattico è riportato a destra) è rappresentato dal termine  $\text{per}(c, \text{div}(c, d))$ .



**VOTO:**

- Esame da 6 CFU, il voto è determinato da questa I parte
- Esame da 9 CFU, è la media pesata della I parte (che vale 2/3) e della II (che vale 1/3) ovvero il voto finale è dato da:  $2 * (\text{votoParteI} + \text{votoParteII}) / 3$  e varia quindi da 0 ad un massimo di 32 (equivalente alla lode).

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 1° parte

10 Gennaio 2013 – Soluzioni

## Esercizio 1

Frasi in *logica dei predicati del I ordine* e loro traduzione in *clausole*:

a. *Tutte le madri amano le loro figlie*

$$\forall X: \text{madre}(X) \wedge \text{figlia}(X, Y) \Rightarrow \text{ama}(X, Y) \\ \{-\text{madre}(X) \vee \neg \text{figlia}(X, Y) \vee \text{ama}(X, Y)\}$$

b. *Grimilde è una madre*

$$\text{madre}(g) \\ \{\text{madre}(g)\}$$

c. *Biancaneve è figlia di Grimilde*

$$\text{figlia}(g, b) \\ \{\text{figlia}(g, b)\}$$

d. *Se una madre ama sua figlia, non l'avvelena*

$$\forall X: \text{madre}(X) \wedge \text{figlia}(Y, X) \wedge \text{ama}(X, Y) \Rightarrow \neg \text{avvelena}(X, Y) \\ \{-\text{madre}(X) \vee \neg \text{figlia}(Y, X) \vee \neg \text{ama}(X, Y) \vee \neg \text{avvelena}(X, Y)\}$$

e. *Grimilde avvelena Biancaneve*

$$\text{avvelena}(g, b) \\ \{\text{avvelena}(g, b)\}$$

*Teoria a clausole:*

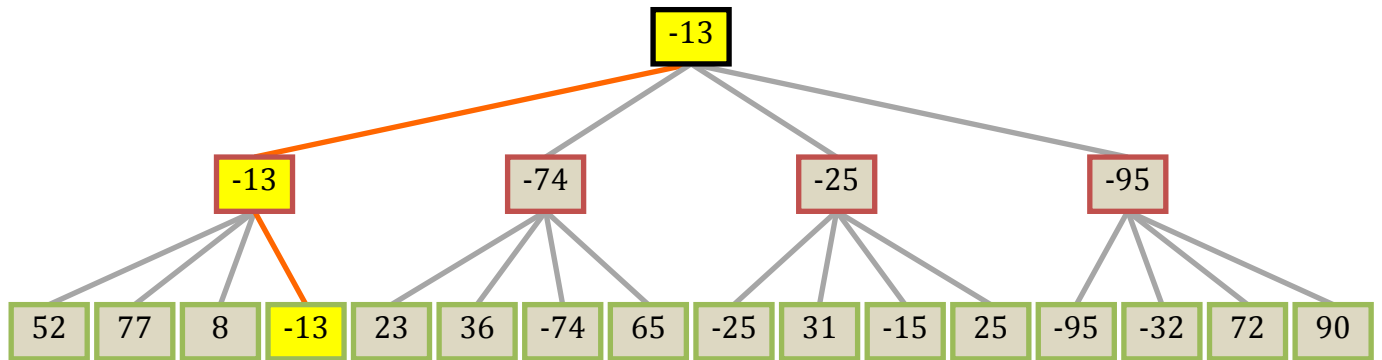
1.  $\neg \text{madre}(X) \vee \neg \text{figlia}(X, Y) \vee \text{ama}(X, Y)$
2.  $\text{madre}(g)$
3.  $\text{figlia}(g, b)$
4.  $\neg \text{madre}(X) \vee \neg \text{figlia}(Y, X) \vee \neg \text{ama}(X, Y) \vee \neg \text{avvelena}(X, Y)$
5.  $\text{avvelena}(g, b)$

*Risoluzione:*

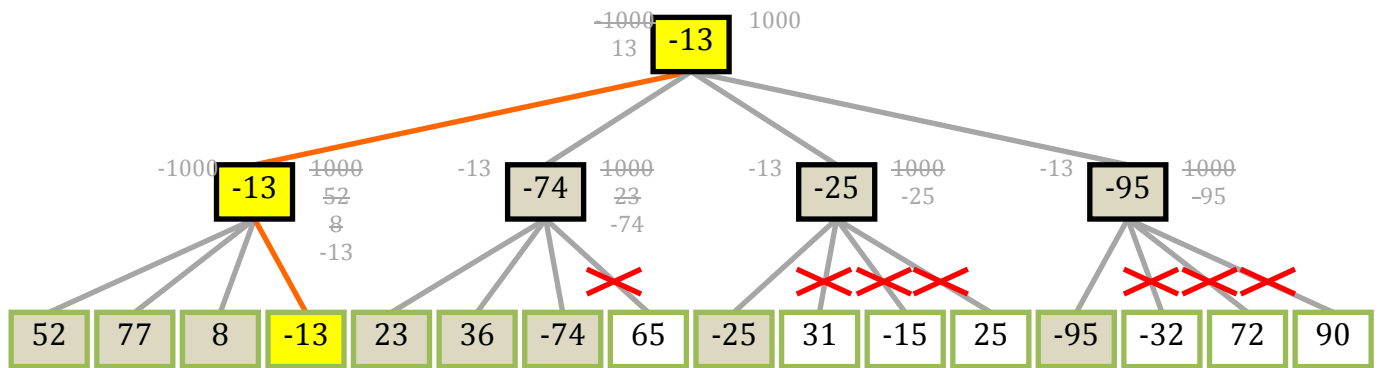
6.  $\neg \text{figlia}(Y, g) \vee \text{ama}(g, Y)$  {X/g}
7.  $\text{ama}(g, b)$  {Y/b}
8.  $\neg \text{figlia}(Y, g) \vee \neg \text{ama}(g, Y) \vee \neg \text{avvelena}(g, Y)$  {X/g}
9.  $\neg \text{ama}(g, b) \vee \neg \text{avvelena}(g, b)$  {Y/b}
10.  $\neg \text{avvelena}(g, b)$
11.  $\square$

## Esercizio 2

min-max:



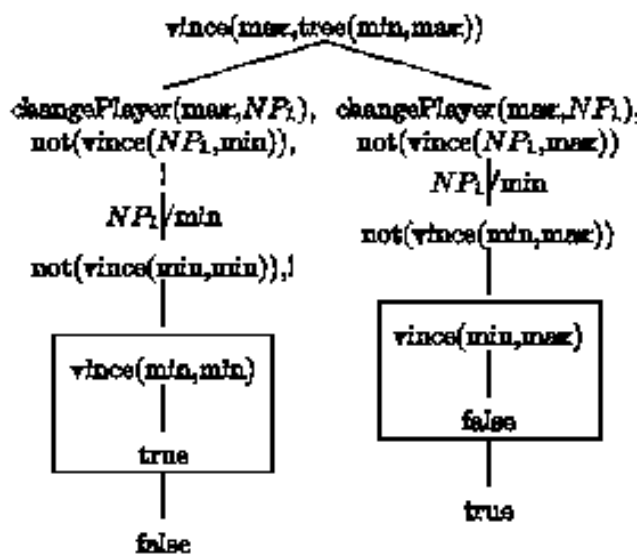
alfa-beta:



I nodi non colorati sono quelli che vengono tagliati nell'algoritmo alfa-beta.

## Esercizio 3

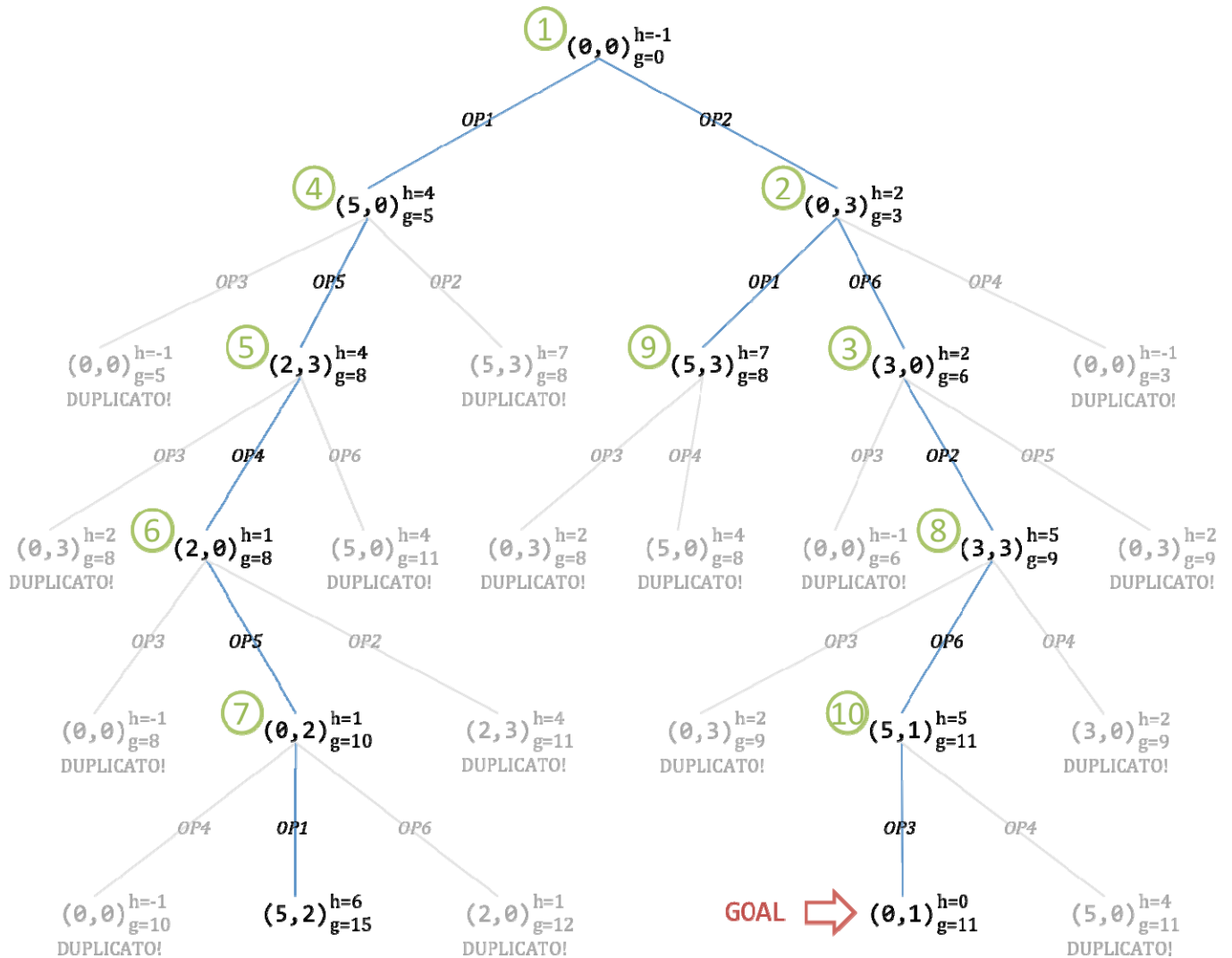
Albero SLD-NF:



## Esercizio 4

```
ultimo(X, [X]).  
ultimo(X, [_|Y]) :- ultimo(X, Y).
```

## Esercizio 5



## Esercizio 6

Vedi slides del corso.

# FONDAMENTI DI INTELLIGENZA ARTIFICIALE – 2° parte

## 10 Gennaio 2013 – Soluzioni

### Esercizio 7

Vedi slides del corso.

### Esercizio 8

```
trova_tutte(Atom, Lout) :-  
    Atom =.. [PredName, X, Y],  
    findall((X,Y), call(Atom), Lout).
```

### Esercizio 9

```
e(per(X, Y)) --> t(X), [ * ], e(Y).  
e(div(X, Y)) --> t(X), [ / ], e(Y).  
e(X) --> t(X).  
t(c) --> [ c ].  
t(d) --> [ d ].
```